

REMARKS

Claim 1 calls for "if not, determining, at the server, why the task was not completed." The task referred to is clearly a task that was not completed that was assigned by the server to one of a network of processor-based client devices. See claim 1, first paragraph. Thus, the net effect of claim 1 is to require that the server determine why a task on the client was not completed. In other words, one device must determine why a task, not executed on that device, was not completed by another device that was supposed to execute that task.

The final rejection relies on a number of references. It remains unclear what reference is believed to teach this element. As will be shown in the following discussion, none of the references teach that element.

It is pointed out in paragraph 6, on page 3 of the final rejection, that Zack allegedly teaches "... the processing controller is capable to detect what causes a task not to be completed, i.e. the current CPU clock setting: See (Abstract:[0057]; [0068]; [0052])." The abstract of Zack has nothing to do with determining at any entity "why the task was not completed." Thus, the abstract does not support the rejection.

Paragraph 57 of Zack talks about a processing controller monitoring the progress of each of the units. It also talks about a prediction of the amount of processing units to complete the task. Again, there is nothing about the processing controller or any other entity determining "why the task was not completed."

Paragraph 68 of Zack talks about the processing controller detecting progress in processing of the task. Again, this has nothing to do with determining "why the task was not completed."

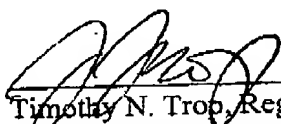
Also cited is paragraph 52 which talks about the processing controller detecting that the current CPU clock setting will not be sufficient to complete processing tasks. The CPU clock setting does not indicate why the task was not completed. The task was not completed within the time and why the task is not completed has nothing to do with the CPU clock setting. That would be like the student telling his teacher the reason he did not finish his homework was that the time he had for the homework elapsed before he got to it. That is not an excuse for why the task was not completed. It is a symptom of non-completion: the time elapsed and the task still was not completed, but that is not why the task was not completed.

In paragraph 8 of the final rejection it is suggested that the applicant's arguments that the remote server could determine why the task was not completed at the client relies on elements not claimed. The claim does not say that the server is "remote" from the client, but I have never heard of a server/client relationship where one is not remote, relative to the other. It is not believed that the word "remote" is in any way material to the argument. The argument is that one entity determines why another entity fails to complete a task. This is not taught by any of the references. The fact that a reference's device may determine itself why it failed to complete the task does not teach the claimed invention. The point is that there are two different entities, one of which determines why the other failed.

Therefore, reconsideration is respectfully requested.

Respectfully submitted,

Date: November 30, 2007



Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
713/468-8880 [Phone]
713/468-8883 [Fax]

Attorneys for Intel Corporation